

(Concise explanations of the disclosure)

Japanese laid-open patent publication No. 2000-181780

Published on June 18, 2000

A semiconductor device may include a data cache or a data cache system and a store queue serving as a write buffer or a store buffer for data-write instruction or data store instruction. Data write operation to a main memory and data cache operation to a data memory may be made, wherein a store instruction including a write address and data is once held by the store queue for improvement in throughput of the processor. Those conventional techniques are disclosed in Japanese laid-open patent publications Nos. 9-114734 entitled "store buffer device", and also in Japanese laid-open patent publications Nos. 2000-181780 entitled "store buffer device". The word "data cache system" is defined to be a data cache system which comprises a tag memory and a data memory.

STORE BUFER DEVICE

Patent Number: JP2000181780
Publication date: 2000-06-30
Inventor(s): YAMAMOTO TAKASHI; KURIHARA TOSHIHIKO
Applicant(s):: HITACHI LTD
Requested Patent: ☐ JP2000181780 (JP00181780)
Application Number: JP19980357479 19981216
Priority Number(s):
IPC Classification: G06F12/00 ; G06F12/08
EC Classification:
Equivalents:

Abstract

PROBLEM TO BE SOLVED: To provide a store buffer device (called as buffer in the following) to a main storage device where merge can be increased and the busy rate of a bus can be suppressed.

SOLUTION: FULL detection logic 24 detects that a buffer 20 becomes full (all entries are valid) by store and timer count logic 23 detects that a store request from a computing element does not exist for not less than time which is previously set. OUT flag (called as a flag in the following) set logic 26 sets a flag to an entry where a VALID bit is erected. When at least one entry where the flag is previously erected exists, flag setting is suppressed. When the flag is set, weep discharge control logic 27 discharges only the entry where the flag is set. Then, new registration in the entry where the flag is not erected is realized during the discharge of the entry from the buffer with the suppression of such flag setting. Then, merge effect is improved with the long residence of store in the buffer.

Data supplied from the **esp@cenet** database - I2

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2000-181780

(P2000-181780A)

(43)公開日 平成12年6月30日(2000.6.30)

(51)Int.Cl.⁷

G 0 6 F 12/00
12/08

識別記号

5 6 0

F I

G 0 6 F 12/00
12/08

テーマコード(参考)

5 6 0 B 5 B 0 0 5
C 5 B 0 6 0

審査請求 未請求 請求項の数4 O L (全 9 頁)

(21)出願番号 特願平10-357479

(22)出願日 平成10年12月16日(1998.12.16)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 山本 敬

神奈川県秦野市堀山下1番地 株式会社日

立製作所汎用コンピュータ事業部内

(72)発明者 栗原 俊彦

神奈川県秦野市堀山下1番地 株式会社日

立製作所汎用コンピュータ事業部内

(74)代理人 100099298

弁理士 伊藤 修 (外1名)

Fターム(参考) 5B005 JJ00 KK12 LL11 MM23 NN02

NN47 PP03 PP11 TT01 VV03

5B060 CB01

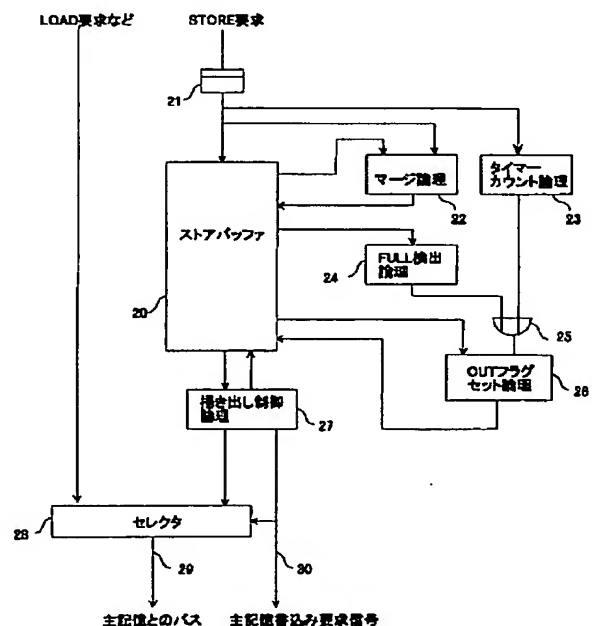
(54)【発明の名称】 ストアバッファ装置

(57)【要約】

【課題】 マージを多くすることができ、バスのビジー率を抑えることが可能な主記憶装置に対するストアバッファ装置(以下、バッファ)の提供。

【解決手段】 FULL検出論理はバッファがストアにより満杯(全エントリが有効)になったことを検出し、タイマーカウンタ論理は演算器からのストア要求が、予め設定された時間以上なかったことを検出し、いずれかの検出により、OUTフラグ(以下、フラグ)セット論理は、VALIDビットの立っているエントリに対しフラグをセットし、既にフラグが立っているエントリが1つでもあればフラグセットを抑止し、フラグがセットされると、掃き出し制御論理はフラグのセットされたエントリのみを掃き出す。前記のようなフラグセットの抑止により、エントリのバッファからの掃き出し中にも、フラグが立っていないエントリへの新規登録を可能とし、バッファでのストアの長期滞留によりマージ効果が上がる。

【図1】



【特許請求の範囲】

【請求項1】 ある処理装置から別の処理装置に対してデータを送る際に、該データを一時的に保持する複数エントリからなるストアバッファ装置において、ストアバッファへの入力が行われてから次にストアバッファへの入力が行なわれるまでの時間を計測する手段と、その計測した時間をあらかじめ設定した設定値と比較する手段とを有し、比較結果に応じてストアバッファからの前記データの掃き出しを開始することを特徴とするストアバッファ装置。

【請求項2】 主記憶装置への書込みアドレスおよびデータ（以下、ストア）を一時保持する複数エントリからなるストアバッファ装置において、該ストアバッファの全エントリに有効なストアが登録されたことを検出し検出信号を出力する検出手段を有し、通常はストアバッファに登録されたストアの取り出しを即時行わずにストアバッファ内に保持し続け、前記検出手段から検出信号が出力された時にストアバッファからのストアの掃き出しを行なうことを特徴とするストアバッファ装置。

【請求項3】 請求項2記載のストアバッファ装置において、ストアバッファへのストアの入力が行われてから次にストアバッファへのストアの入力が行なわれるまでの時間を計測する手段と、その計測した時間をあらかじめ設定した設定値と比較する手段とを持ち、比較結果に応じてストアバッファからのストアの掃き出しを行なうことを特徴とするストアバッファ装置。

【請求項4】 請求項2または請求項3記載のストアバッファ装置において、ストアバッファからの掃き出しの対象となったエントリにマークを付け、該マークが既に一つでも付いている間は新たに他のエントリに該マークを付けることを抑止する手段を有し、該マークのついたエントリのストアのみをストアバッファから掃き出し、該マークの付いたエントリのストアをストアバッファから掃き出している間にも、該マークのついていないエントリへのストアの新規登録を可能とすることを特徴とするストアバッファ装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、演算装置から主記憶装置へのストアアドレスおよびデータを一時保持する複数エントリからなるストアバッファ装置に係り、特に、効率的な利用を可能とするストアバッファ装置に関する。

【0002】

【従来の技術】今日の情報処理装置の多くは、主記憶装置ならびにキャッシュメモリへのデータ書込みの際、書込みアドレスおよびデータを一旦保持するためのストア

バッファを持っている。これは、ロード命令のレイテンシ（データを読むまでの待ち時間）が、プログラムの処理性能に大きく影響するのに対して、ストア命令のレイテンシ（データが主記憶やキャッシュに反映されるまでの時間）は実際の処理性能にはほとんど影響しないため、ロードを優先してストアを後回しにするという目的で利用されている。

【0003】従来、主記憶装置に対する書込み用のストアバッファでは、一旦ストア情報をストアバッファに登録し、主記憶との間のバスの空き具合を見て、ストアバッファより優先度の高いアクセス要求が無ければストアバッファから取り出して主記憶に送るという制御方式が一般的である。この例は、特開平8-221311や特開平5-181742などに記載されている。また、ストアバッファへの登録時にはマージ処理を行なうのが一般的である。マージ処理とは、既にストアバッファに登録されているストアに対し、同一アドレスのストアが後続して行われた場合、先行ストアの入っているエントリにその後続のストアを上書きする（新しいエントリを使用しない）ことで主記憶に対するストア情報の数を減らす処理のことである。このストアバッファのマージ機構は、先行ストアが長い間ストアバッファ内に滞留するほど、またなるべく多くのストアがストアバッファ内にあるほど、マージの機会が増えて有効に働くことになる。

【0004】しかしながら、主記憶装置に対する書込み用のストアバッファでは、上述したように競合するアクセスがなければすぐに掃き出しを行ってしまう方式が一般的であり、そのためストアバッファ内に滞留する時間が平均的には非常に短いため、このマージ機構が有効に働かないことが多い。例えば、特開平5-181742ではこの点を指摘し、連続する2つのストアに限ってマージを行うという方式が示されているが、やはりマージ可能ケースがかなり限定されてしまうため、様々なメモリアクセスパターンに対して有効とは言い難い。

【0005】さらに、従来の制御方式では、バスが空き次第ストア情報がストアバッファから掃き出されてしまうため、ストア情報の掃き出しがまばらになり、主記憶との間のデータバスで、主記憶からのReadデータとプロセッサからのWriteデータが入り交じることになる。つまり、ReadとWriteの方向切り替えが頻繁に発生することになるが、この方向切り替えの時間は無駄なサイクルとなるため、バスの使用効率が落ち、全体性能を低下させる原因ともなりうる。

【0006】

【発明が解決しようとする課題】上述したように、従来方式によるストアバッファ装置では、マージ機構が有効に働かないという問題点がある。さらに、ストアバッファからの掃き出しがまばらになるため、主記憶とのバス上でRead/Write切り替えが頻発し、性能低下の要因ともなりうる。

【0007】逆に、たとえばストアバッファの掃き出しを積極的に行わず、まとめて掃き出すような方式を用いた場合、ストアデータが主記憶に反映されるまでの時間が問題となる。つまり、ストアバッファ内にストア情報をなるべく滞留させるような制御を行なった場合、そのストア情報が主記憶にいつ反映されるかがまったく保証されないため、最悪のケースでは永久にストアバッファから掃き出されないようなケースも発生しうる。例えば、共有メモリのマルチプロセッサ構成のシステムにおいて、お互いのプロセッサのストアした結果を相手のプロセッサが待ち、お互いの書きこんだ値を確認してから次の動作に移るような処理を行なう場合、ストア情報が主記憶に反映されるまでは相手のストアした値が読めないため、お互いに待ち続けるが、待っている間は後続のストア命令が降ってこないためにストアバッファが満杯にならず、ストアバッファの掃き出しが開始されないため、いつまで待ってもお互いのストアデータが読めないという状況に陥る。

【0008】さらに、ストアバッファをまとめて掃き出すような方式では、掃き出しを開始した後に新しくストアバッファに送られてきたストア情報の扱いが難しくなる。従来は、ストアバッファ掃き出し中は新たなエントリを登録しないという単純な方式が取られるのが一般的であったが、これだと掃き出しのたびにストア情報の受け付けが止まり、演算器側の性能が悪くなるという問題点がある。一方、掃き出し中にもストアバッファへの登録を可能にした場合、掃き出ししている最中にエントリが増えていくため、それらも続けて掃き出さざるを得ないが、その場合、ストアバッファの掃き出しがかなりの長時間にわたって主記憶とのバスを占有してしまう可能性があり、本来優先すべきLOAD系の性能が低下してしまうという問題がある。

【0009】本発明は、上述のような事情に鑑みてなされたものであり、ストアバッファのマージ機能や複数のストアをまとめて掃き出す機能を生かしつつ、上記した不具合を回避し、性能向上を図ったストアバッファ装置を提供することを目的とする。

【0010】

【課題を解決するための手段】上記目的を達成するため、本発明は、ある処理装置から別の処理装置に対してデータを送る際に、該データを一時的に保持する複数エントリからなるストアバッファ装置において、ストアバッファへの入力が行われてから次にストアバッファへの入力が行なわれるまでの時間を計測する手段と、その計測した時間をあらかじめ設定した設定値と比較する手段とを有し、比較結果に応じてストアバッファからの前記データの掃き出しを開始するようにしている。

【0011】また、主記憶装置への書き込みアドレスおよびデータ（以下、ストア）を一時保持する複数エントリからなるストアバッファ装置において、該ストアバッファ

の全エントリに有効なストアが登録されたことを検出し検出信号を出力する検出手段を有し、通常はストアバッファに登録されたストアの取り出しを即時行わずにストアバッファ内に保持し続け、前記検出手段から検出信号が出力された時にストアバッファからのストアの掃き出しを行なうようにしている。

【0012】また、上記検出手段を有するストアバッファ装置において、ストアバッファへのストアの入力が行われてから次にストアバッファへのストアの入力が行なわれるまでの時間を計測する手段と、その計測した時間をあらかじめ設定した設定値と比較する手段とを持ち、比較結果に応じてストアバッファからのストアの掃き出しを行なうようにしている。

【0013】また、ストアバッファからの掃き出しの対象となったエントリにマークを付け、該マークが既に一つでも付いている間は新たに他のエントリに該マークを付けることを抑止する手段を有し、該マークのついたエントリのストアのみをストアバッファから掃き出し、該マークの付いたエントリのストアをストアバッファから掃き出ししている間にも、該マークのついていないエントリへのストアの新規登録を可能とするようにしている。

【0014】

【発明の実施の形態】以下、本発明によるストアバッファ(STB)装置の実施例を図面により詳細に説明する。図1は、本発明の一実施例によるストアバッファ装置の構成を示すブロック図である。20はストアバッファ、21はラッチ、22はマージ論理、23タイマカウンタ論理、24はFULL検出論理、25はオアゲート、26はOUTフラグセット論理、27は掃き出し制御論理、28はセレクトである。演算器から送られてきた主記憶に対するストア要求は、ラッチ21で受け取った後、マージの可否のチェックが行なわれ、ストアバッファ本体20へ登録される。ストアバッファ(STB)装置を構成する上記論理等についての説明、および、ストアバッファ装置の動作説明を、以下に順次行う。

【0015】ストアバッファ本体20の構成を図2、マージ論理22の構成を図3に示す。図2では、複数エントリで構成されるストアバッファの1エントリ分の構成を示している。VALIDビット40は、そのエントリに有効なストア情報が登録されていることを示す。ストア情報がストアバッファに登録されるとVALIDビット40は'1'になる。OUTフラグ41は、そのエントリがストアバッファからの掃き出し対象になったことを示すが、登録の初期状態では'0'がセットされる。残りのアドレスとデータのフィールドには、演算器から送られてきたストア情報（以下、ストアという）のアドレスとデータが格納される。

【0016】図3は、マージのチェックを行い、ストアバッファ内の登録先エントリを決定するマージ論理22の詳細構成を示している。ストアバッファに対し、新規ス

トアの登録要求が来ると、各エントリに登録されているストアアドレスと、新しく登録しようとしているストアのアドレスとを比較器53で比較するが、これと同時に各エントリのVALIDビットとOUTフラグをチェックし、「エントリが有効かつストアバッファからの掃き出し対象になっていない」ことをANDゲート52にて検査する。ANDゲート52での検査結果と比較器53での比較結果がともに'1'であった場合にのみ、ANDゲート54の出力が'1'となり、そのエントリにマージ可能であることを示す(ここで、OUTフラグの立っているエントリをマージ対象から外すことで、掃き出し中(あるいは掃き出し直後)のエントリに誤ってマージしてしまうことを防いでいる)。このチェックを、ストアバッファの全エントリについて同時に行ない、どれかのエントリとマージ可能である場合にはORゲート56の出力が'1'となるため、新規ストアの登録先エントリのポインタ(セクタ57の出力)は、そのマージ可能エントリのエントリ番号となる。一方、空きエントリポインタ50は、未登録のストアバッファエントリ番号を保持しており、もしどのエントリともマージできない場合(ORゲート56の出力が'0'の場合)は、セクタ57の出力にはこの空きエントリポインタが選ばれ、この場合、新規ストアはまだ何も登録されていない新しいエントリに登録され、空きエントリポインタ50はインクリメント(+1)される。

【0017】ストアバッファに登録されたストアは、従来技術では、主記憶とのバスを使用する他の要求とのプライオリティをとった後すぐに掃き出されていたが、本発明に基づく本実施例の場合、積極的には掃き出しを行わず、(1)バッファが満杯(バッファフル)のケースと、(2)一定時間新たなストア要求が無いケースのいずれかで掃き出しが行われる。

【0018】図1におけるタイマーカウント論理23の構成を図4に、FULL検出論理24の構成を図5に示す。図4に示すタイマーカウント論理23では、あらかじめ設定した時間以上の間、演算器からの新たなストア要求が来ないことを検出している。具体的には、ラッチ72に初期値'0'を設定した状態でSTB入力ラッチ21のValid信号75(ストア要求があったことを示す)を監視する。このValid信号75が'1'(ストア要求があった)のときにはラッチ72には'0'が入力されるが、Valid信号75が'0'(ストア要求が無かった)の時にはラッチ72には前の値に1加算した値が入力され、これが繰り返されることで、ストア要求が無かった時間を計測する。ラッチ73にはあらかじめ任意のタイムアウト値(固定値)を入力しておき、ラッチ72で計測した値がこのラッチ73に設定した値と一致すると、比較器74の出力としてOUTフラグセット要求信号が出力される。後述するが、これはストアバッファの掃き出しを行なうために用いられる。一方、図5に示すFULL検出論理では、ストアバッファ各エントリのVALIDビットを監視し、すべてのエントリに有効なストアが登録さ

れた時、ANDゲート80の出力としてOUTフラグセット要求信号が出力される。

【0019】図1において、これらのOUTフラグセット要求信号は、ORゲート25に入力され、その出力がOUTフラグセット論理26に入力される。このOUTフラグセット論理によってストアバッファのOUTフラグがセットされるが、それによって掃き出し制御論理27が起動され、ストアバッファの掃き出しが行われる。

【0020】図6に、OUTフラグセット論理26の構成を示す。OUTフラグセット要求信号が入力されると、VALIDビットの立っているエントリに対してOUTフラグのセットを行なう。この時、ORゲート90で各エントリのOUTフラグを監視し、どこかのエントリにOUTフラグが既に立っている場合にはインバータ91の出力が'0'になって、OUTフラグのセットを抑止する。つまり、一度OUTフラグが立つと、それらのエントリが全部掃き出されるまで次のOUTフラグは立てられない。

【0021】図7に、掃き出し制御論理の構成を示す。STBからの取り出しポインタ106は、主記憶に書き込むアドレス・データを保持するSTBエントリ番号を選択するためのポインタである。このポインタによって、STBのある1エントリが選択され、その中のVALIDビットとOUTフラグが検査される。そして、両方とも'1'であった場合、主記憶に対する書き込み要求信号107が生成され、それと同時にセクタ100で選択されたアドレス・データが送出される。主記憶に対する書き込み要求信号107は、セクタ105にも送られ、この際にSTBからの取り出しポインタ106がインクリメント104によって+1される。逆に、OUTフラグが立っていないケースでは、主記憶書き込み要求信号107が生成されず、STBからの取り出しポインタ106も更新されない。つまり、先述したOUTフラグセット論理26によってOUTフラグがセットされない限り、主記憶への書き込み要求は行われないことになる。

【0022】これらの機構により、図1に示す本実施例のストアバッファ装置では、以下のような動作を行なう。まず、演算器からのストア要求があると、それはラッチ21でマージ可否の判定が行なわれ、ストアバッファに登録される。この時、既に登録されているエントリとマージ可能と判断された場合には、そのエントリに上書きされ、そうでない場合には新しいエントリに登録される。演算器からのストア要求がある設定された時間以上行われない場合、もしくはストアバッファが満杯になった場合、タイマーカウント論理23またはFULL検出論理24によってそれが検出され、OUTフラグセット論理26によって、登録されているストアバッファエントリのOUTフラグが一斉にセットされる。OUTフラグがセットされると、掃き出し制御論理27によってストアバッファからの掃き出しが開始される。

【0023】掃き出し動作を行なっている最中も、演算器からのストア要求を受け付けてストアバッファに登録

することが出来るが、OUTフラグは立たない(先に立てたOUTフラグが1エントリでも残っているうちは次のOUTフラグは立てられない)ので、掃き出し動作中に新規登録したエントリがすぐに掃き出されてしまうことはない。逆に、次の掃き出し条件が整うまではストアバッファ内に滞留するため、マージの機会が増える可能性がある。また、OUTフラグが立っているエントリはマージ動作の対象とならないため、掃き出し中(あるいは掃き出し直後)のエントリに誤ってマージを行なってしまうようなことはない。

【0024】ストアが掃き出されたエントリは、VALIDビットとOUTフラグが'0'に落され、新規登録が可能なエントリとなる。これにより、ストアバッファが満杯になっても掃き出しを開始すればすぐに空きが出来るため、演算器からのストア要求を止めている時間を最小限にすることが出来る(従来のストアバッファ装置では、掃き出し動作中は演算器からのストア要求は一切受け付けられないという方式が一般的である)。さらに、ストアバッファから連続して掃き出されるストアの数の上限は、同時にOUTフラグを立てられる数、すなわちストアバッファのエントリ数となるため、主記憶とのインタフェースをストアのみで独占して他のロード命令等の性能を落してしまうことが避けられ、また、連続して受け取るべきストアの数があらかじめ分かっていたら、ストアを受け取る主記憶側の論理が作成しやすいという利点もある。

【0025】

【発明の効果】以上説明したように、本発明によれば、ストアバッファへの登録要求がある一定時間以上発生しない場合に、ストアバッファから強制的に掃き出しを行なうことで、通常はストアバッファ内にストアを滞留させてストアバッファでのマージ処理を有効に行ないつつ、主記憶に対してデータを確実に反映することが出来る。また、ストアバッファの掃き出し対象のエントリにマーク(OUTフラグ)を付加することで、ストアバッファの掃き出しを行ないながら、新規ストアを正しく新規エントリに登録することができ、さらにOUTフラグが仕切りとなって、一度に掃き出しを行なう数を制限することができる。

【図面の簡単な説明】

【図1】本発明の一実施例によるストアバッファ装置の構成を示すブロック図である。

【図2】図1のストアバッファの1エントリ分の構成例を示す図である。

【図3】図1のマージ論理の構成例を示す図である。

【図4】図1のタイマーカウント論理の構成例を示す図である。

【図5】図1のFULL検出論理の構成例を示す図である。

【図6】図1のOUTフラグセット論理の構成例を示す図である。

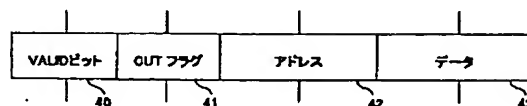
【図7】図1の掃き出し制御論理の構成例を示す図である。

【符号の説明】

- 20 スタアバッファ
- 21 ラッチ
- 22 マージ論理
- 23 タイマーカウント論理
- 24 FULL検出論理
- 25 OR回路
- 26 OUTフラグセット論理
- 27 掃き出し制御論理
- 28 セレクタ
- 50 空きエントリポインタ
- 51 インバータ(NOT回路)
- 52, 54 AND回路
- 53 比較器
- 55 エンコーダ
- 56 OR回路
- 57 セレクタ
- 70 インクリメンタ
- 71 セレクタ
- 72, 73 ラッチ
- 74 比較器
- 75 STB入力ラッチのVALID(有効)信号
- 80 AND回路
- 90 OR回路
- 91 インバータ(NOT回路)
- 92, 93, 94 AND回路
- 100, 101, 102, 105 セレクタ
- 103 AND回路
- 104 インクリメンタ
- 106 ラッチ

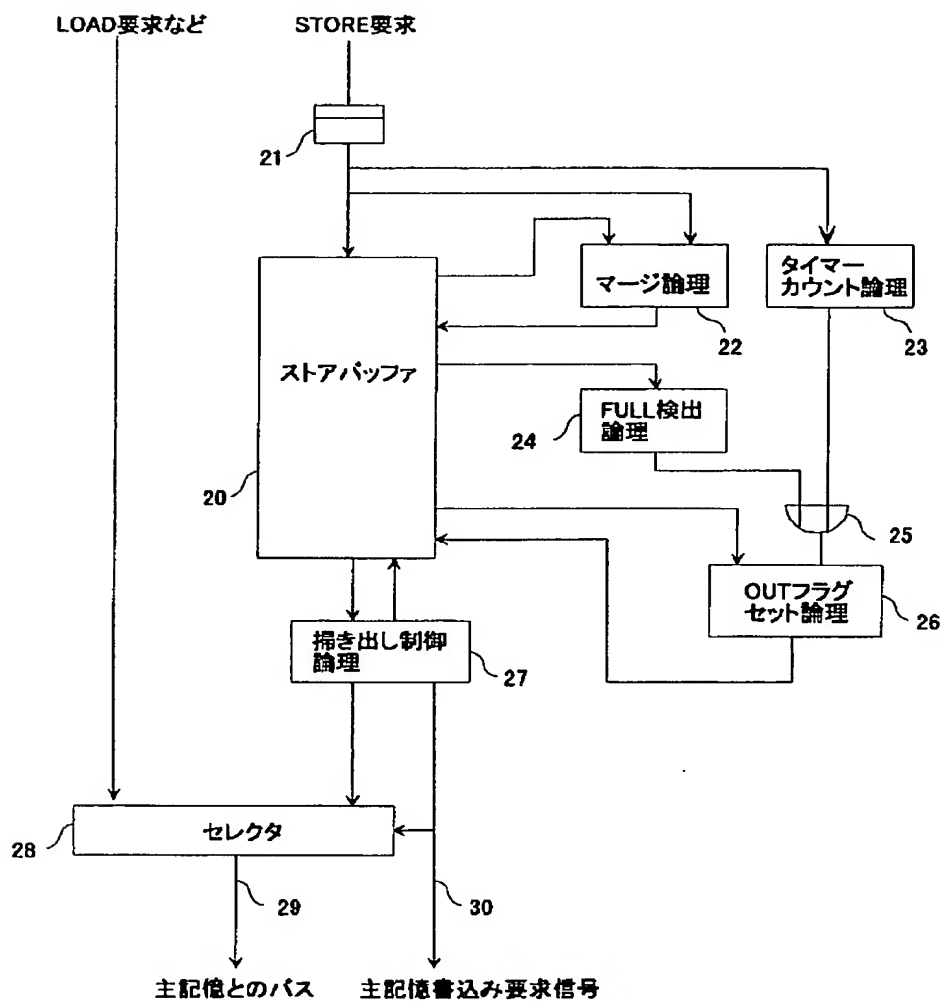
【図2】

【図2】



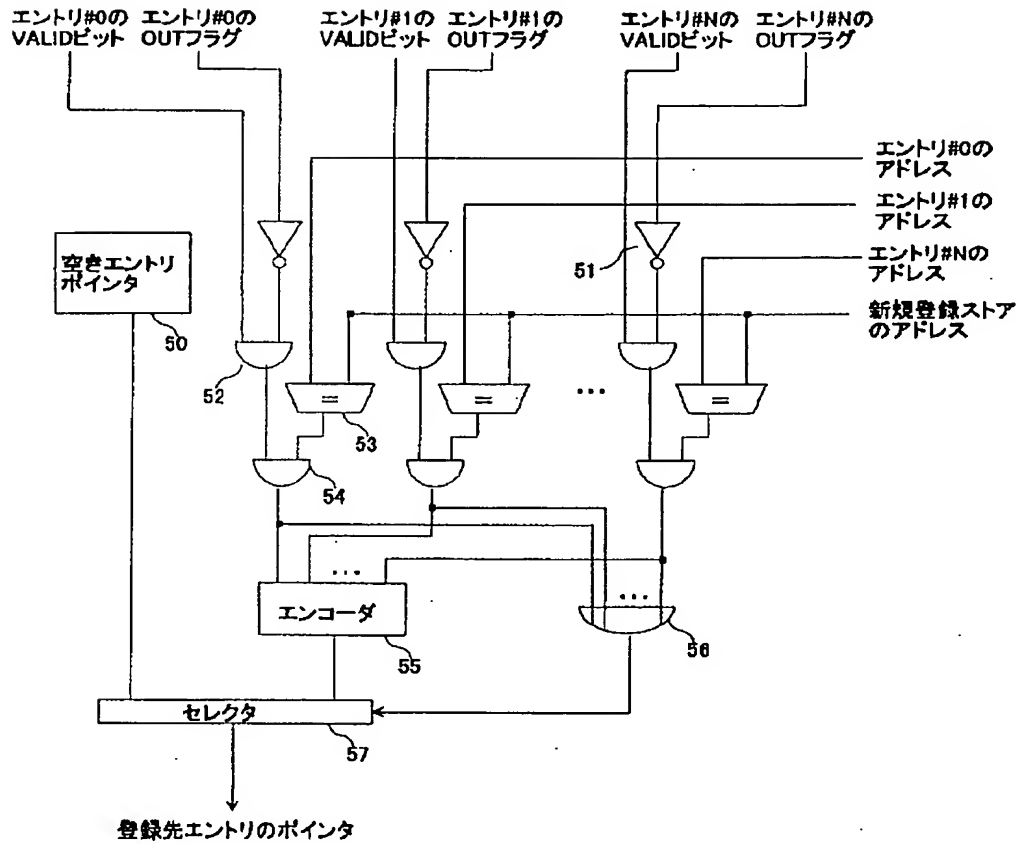
【図1】

【図1】



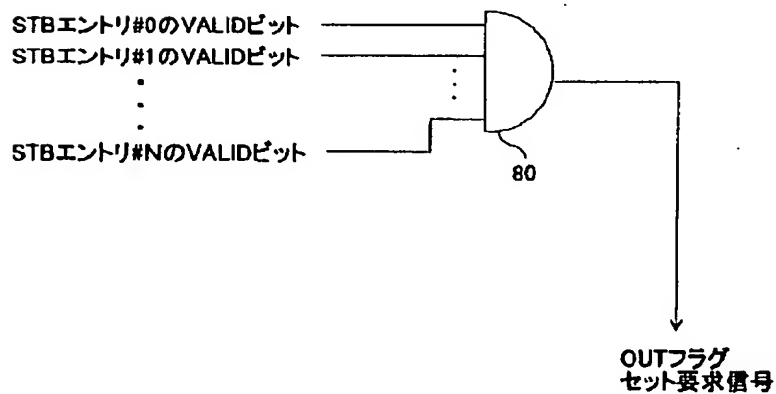
【図3】

【図3】



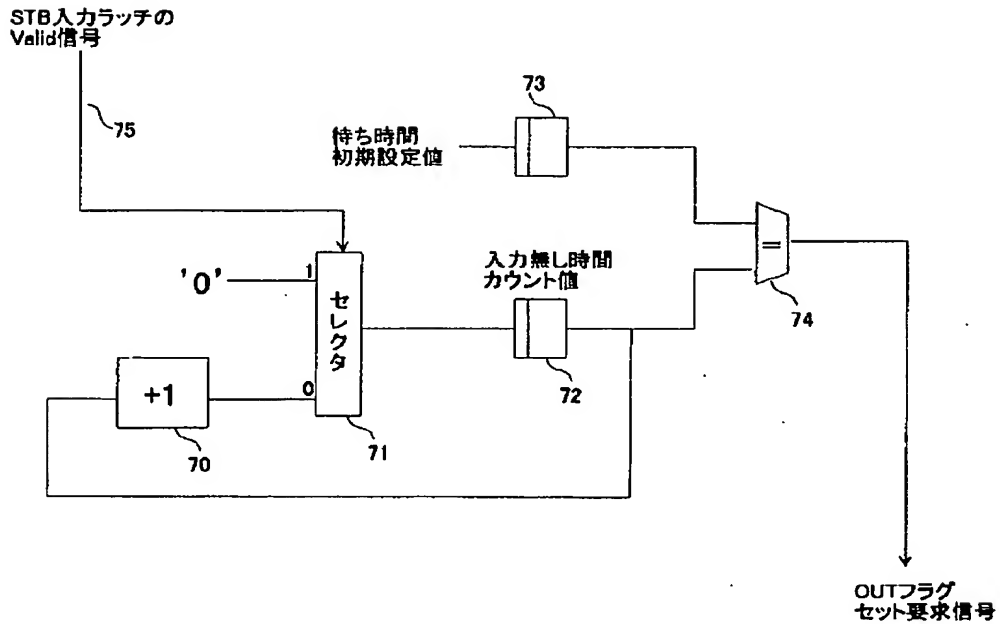
【図5】

【図5】



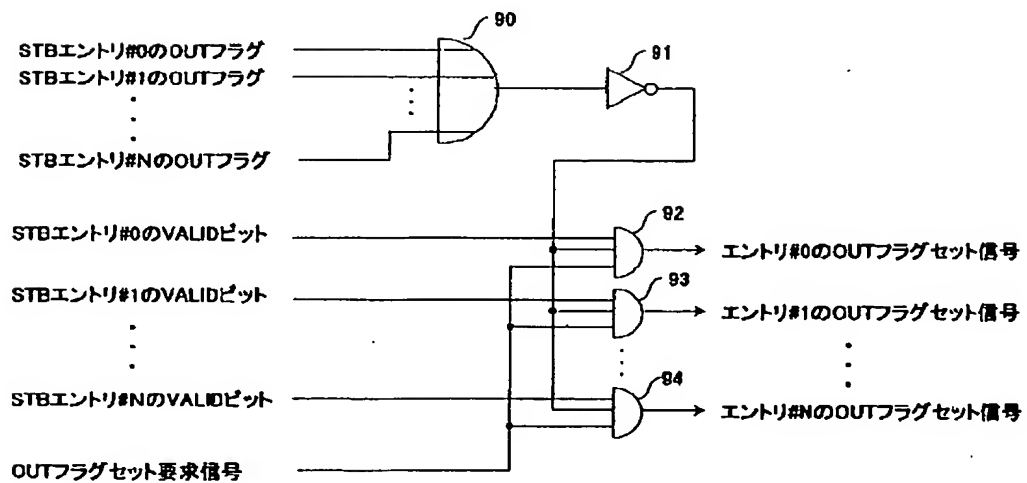
【図4】

〔図4〕



【図6】

〔図6〕



【図7】

【図7】

